Route Models

Chargement : model hook

Une route est responsable du chargement du ou des models qui seront affichés dans le template associé.

Soit le fichier de routage suivant :

```
Router.map(function() {
  this.route('users');
});
```

Le chargement de models est réalisé dans la fonction model() (model hook) du router handler associé :

```
import Route from '@ember/routing/route';

export default Route.extend({
  model() {
    return this.get('store').findAll('user');
  }
});
```

Généralement, le **model hook** retourne une instance **Ember Data**, mais il peut également retourner un objet **promise** (requête asynchrone par exemple), un objet javascript ou un tableau.

```
import Route from '@ember/routing/route';

export default Route.extend({
  model() {
    return [
        {name: 'Higgins', age: 40},
        {name: 'Rosario', age: 20},
        {name: 'Travis', age: 30}
        ];
  }
});
```

La valeur retournée par le model hook de la route est ensuite affectée à la propriété model du controller (implicite ou explicite). Il est ensuite possible d'utiliser le model dans le template associé.

```
<h1>Utilisateurs</h1>
{{#each model as |user|}}
  <p>{{user.name}}</p>
{{/each}}
```

Models dynamiques

Le ou les models chargés dépendent parfois des interactions avec l'utilisateur.

Avec Ember, cette fonctionnalité est mise en place avec les routes dynamiques.

```
Router.map(function() {
  this.route('user', { path: '/users/:user_id' });
});
```

```
import Route from '@ember/routing/route';

export default Route.extend({
  model(params) {
    return this.get('store').findRecord('user', params.user_id);
  }
});
```

Models multiples

Dans certains cas, la route peut avoir besoin de plusieurs models. Ember utilise dans ce cas la méthode hash du package RSVP permettant de définir une seule **promise** à partir de plusieurs **promises** existantes :

```
import Route from '@ember/routing/route';
import RSVP from 'rsvp';

export default Route.extend({
  model() {
    return RSVP.hash({
      songs: this.get('store').findAll('song'),
      albums: this.get('store').findAll('album')
    });
  }
});
```

Contexte parent

Il arrive qu'une route ne possède pas les paramètres d'url dont elle aurait besoin, en particulier s'il s'agit d'une route enfant.

Il est possible dans ce cas d'utiliser la méthode **paramsFor** pour récupérer les paramètres adéquats d'une des routes parent.

Dans le cas suivant, la route **album/index** récupère les paramètres de la route **album** :

```
import Route from '@ember/routing/route';
```

```
export default Route.extend({
  model() {
    let { album_id } = this.paramsFor('album');

    return this.store.query('song', { album: album_id });
  }
});
```

Il est également possible de récupérer le model délivré par une route parent avec la méthode **modelFor** :

```
import Route from '@ember/routing/route';

export default Route.extend({
  model() {
    let { songs } = this.modelFor('album');

    return songs;
  }
});
```

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/richclient/emberjs/router/models

Last update: **2019/08/31 14:21**